

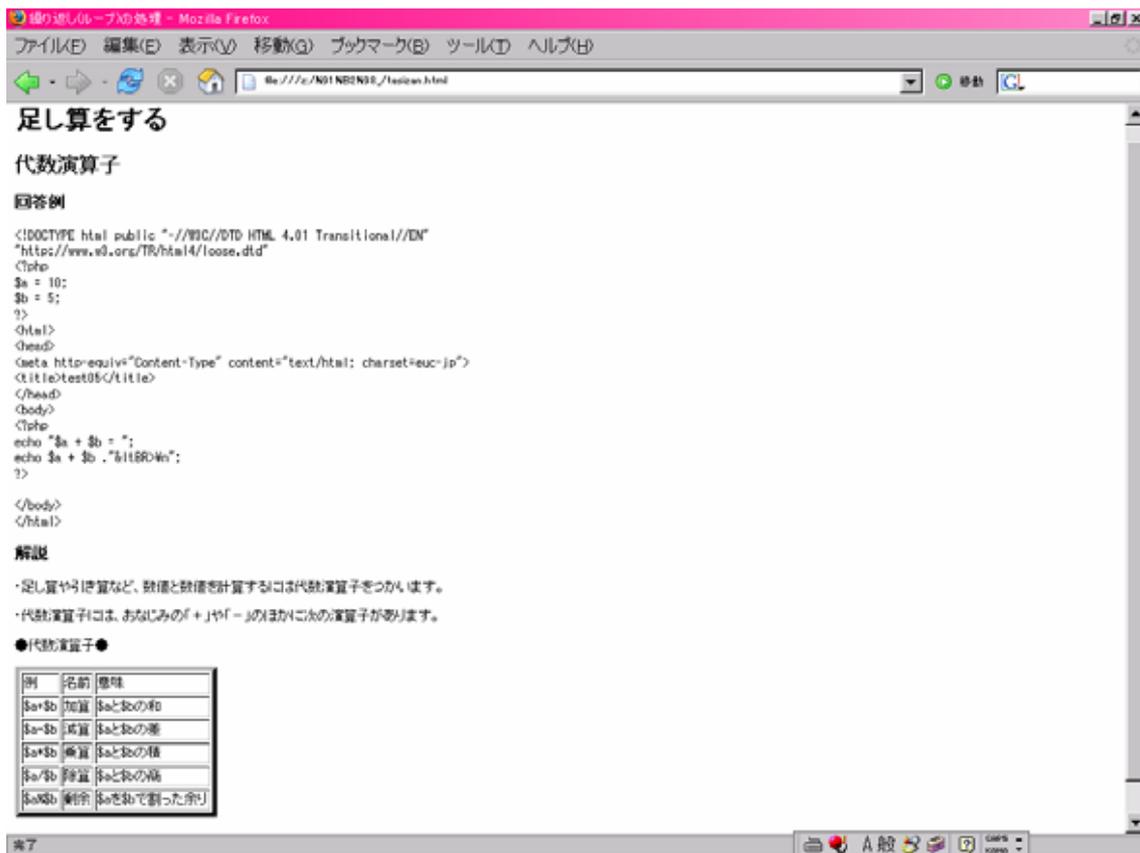
図 1



図 4



図 7



プログラミングの第1歩である、文字の表示を試みましょう。まずはブラウザへ「hello!」と表示してみます。次のスクリプトを実行してみましょう。

Mojiretsu.html

```
001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <?php
003: $mondai1 = nl2br($_POST['mondai1'])
004: ?>
005: <html lang="ja">
006: <head>
007: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
008: <meta http-equiv="Content-Style-Type" content="text/css">
009: <title>文字列を表示する</title>
010: </head>
011: <h1>文字列を表示する</h1>
012: <h2>「hello!」の文字を表示します。</h2>
013: <h3>回答例</h3>
014: <body>
015: <?php
016: echo $mondai1;
017: ?>
018: <pre>
019: &lt;!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"
020: "http://www.w3.org/TR/html4/loose.dtd"
021: &lt;html&gt;
022: &lt;head&gt;
023: &lt;meta http-equiv="Content-Type" content="text/html; charset=euc-jp"&gt;
024: &lt;title&gt;test01&lt;/title&gt;
025: &lt;/head&gt;
026: &lt;body&gt;
027: &lt;?php
028: echo ("hello!");
029: ?&gt;
030: &lt;/body&gt;
031: &lt;/html&gt;
032: </pre>
033:
034: <h3>解説</h3>
035: <p>・ echo ("hello!");という一行が「hello!」という文字列になって表示されています。
036: この「echo」は、文字を表示するという命令です。echoのような命令のことを関数といいます。
037: また、echo 関数の後ろにある"hello!"の部分を変数といいます。</p>
038: <p>・ &lt;?php~?&gt; を PHP の初めと終わりにつけることで、その部分が PHP スクリプトであることを示しています。</p>
039: <p>・ 文章の終わりに「。」をつけるように、PHP では文の終わりに必ず「;」をつけなければなりません。
040: ;をつけないと文の終わりとみなされないのが、途中で改行が入っても問題はありません。
041: 次のように記述しても問題はありませんが、分かりにくいので一般的には一行に一文を記述します。</p>
042: <pre>&lt;?php
043: echo "hell
044:     o!";
045: echo "こん
046:     に
047:     ちは!";
048: ?&gt;</pre>
049:
050: <h3><a href="sotsuron.html">TOP</a></h3>
051: </body>
052: </html>
```

名前とともに現在時刻を表示させましょう。date 関数により、現在時刻を取得することができます。

Nitiji.html

```
001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <html lang="ja">
003: <head>
004: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
005: <meta http-equiv="Content-Style-Type" content="text/css">
006: <title>現在の日時を表示する</title>
007: </head>
008: <h1>現在の日時を表示する</h1>
009: <h2>現在の日にちと時間を表示します。</h2>
010: <h3>回答例</h3>
011: <body>
012:
013: <pre>
014: &lt;!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"
015: "http://www.w3.org/TR/html4/loose.dtd"
016: &lt;html&gt;
017: &lt;head&gt;
018: &lt;meta http-equiv="Content-Type" content="text/html; charset=euc-jp"&gt;
019: &lt;title&gt;test02&lt;/title&gt;
020: &lt;/head&gt;
021: &lt;body&gt;
022: &lt;?php
023: $datetime = date("Y年m月d日 G:i:s");
024: echo ("現在の日時は $datetime");
025: ?&gt;
026: &lt;/body&gt;
027: &lt;/html&gt;
028: </pre>
029:
030: <h3>解説</h3>
031: <p>・日時を表示したいときは、$datetime を使用します。</p>
032: <p>・ここで使われている=date の後の Y, m, d, G, i, s はタイムスタンプです。</p>
033: <p>●タイムスタンプで使用する文字●</p>
034: <table border=4 width=500 >
035: <tr>
036: <td>a</td>
037: <td>"am"または"pm"を表します。</td>
038: </tr>
039: <tr>
040: <td>A</td>
041: <td>"AM"または"PM"を表します。</td>
042: </tr>
043: <tr>
044: <td>y</td>
045: <td>70 や 99 のような 2 桁の数字による"年"を表します。</td>
046: </tr>
047: <tr>
048: <td>Y</td>
049: <td>1980 や 2001 のような 4 桁の数字による"年"を表します。</td>
050: </tr>
051: <tr>
052: <td>d</td>
053: <td>05 や 22 などのような 2 桁の数字による"日"を表します。値は必ず 2 桁になるので、値の範囲は"01"から
"31"になります。(先頭に 0
054: を付けたくない時は"j"を使う)</td>
```

```

055: </tr>
056: <tr>
057: <td>D</td>
058: <td>Sun や Fri のような 3 文字のテキスト形式による“曜日”を表します。</td>
059: </tr>
060: <tr>
061: <td>l</td>
062: <td>Sunday や Friday のような長いテキスト形式の“曜日”を表します。</td>
063: </tr>
064: <tr>
065: <td>F</td>
066: <td>January や June のような長いテキスト形式の“月”を表します。</td>
067: </tr>
068: <tr>
069: <td>m</td>
070: <td>04 や 10 のような 2 桁の数字による、“月”を表します。値は必ず 2 桁になるので、値の範囲は“01”から
“12”になります。(先頭に
071: 0 を付けたくない時は“n”を使う。)</td>
072: </tr>
073: <tr>
074: <td>M</td>
075: <td>Jan や Oct のような 3 文字のテキスト形式による“月”を表します。</td>
076: </tr>
077: <tr>
078: <td>h</td>
079: <td>05 や 11 のような 2 桁の数字による 12 時間単位の“時”を表します。値は必ず 2 桁になるので、値の範囲
は“00”から“12”になります。(先頭に
080: 0 を付けたくない時は“g”を使う。)</td>
081: </tr>
082: <tr>
083: <td>H</td>
084: <td>09 や 20 のような 2 桁の数字による 24 時間単位の“時”を表します。値は必ず 2 桁になるので、値の範囲
は“00”から“23”になります。(先頭に
085: 0 を付けたくない時は“G”を使う。)</td>
086: </tr>
087: <tr>
088: <td>i</td>
089: <td>05 や 35 のような 2 桁の数字による“分”を表します。値は必ず 2 桁になるので、値の範囲は“00”から“59”
になります。</td>
090: </tr>
091: <tr>
092: <td>s</td>
093: <td>09 や 52 のような 2 桁の数字による“秒”を表します。値は必ず 2 桁になるので、値の範囲は“00”から“59”
になります。</td>
094: </tr>
095: <tr>
096: <td>U</td>
097: <td>Unix 基準時(1970 年 1 月 1 日 00:00:00GMT)からの秒数を表します。</td>
098: </tr>
099: </table>
100:
101: <h3><a href="sotsuron.html">TOP</a></h3>
102: </body>
103: </html>

```

クライアントのホスト名と現在時刻を表示させ、現在時刻によって違ったメッセージを表示させてみます。

Kankyuu.html

```
001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <html lang="ja">
003: <head>
004: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
005: <meta http-equiv="Content-Style-Type" content="text/css">
006: <title>環境変数を表示する</title>
007: </head>
008: <h1>環境変数を表示する</h1>
009: <h2>環境変数の取得方法と表示方法</h2>
010: <h3>回答例</h3>
011: <body>
012: <p>●クライアント情報の表示●</p>
013: <pre>
014: &lt;!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"
015: "http://www.w3.org/TR/html4/loose.dtd"
016: &lt;?php
017: $name = "クライアントの名前";
018: $text = "$name さん、ようこそ!";
019: $datetime = date("Y年m月d日 G:i:s");
020: ?&gt;
021: &lt;html&gt;
022: &lt;head&gt;
023: &lt;meta http-equiv="Content-Type" content="text/html; charset=euc-jp"&gt;
024: &lt;title&gt;test03&lt;/title&gt;
025: &lt;/head&gt;
026: &lt;body&gt;
027: &lt;?php
028: echo $datetime;
029: echo "<BR>\n";
030:
031: if ($hour >= 5 && $hour < 12) {
032:   echo "おはようございます" ;
033: } elseif ($hour >= 12 && $hour < 18) {
034:   echo "こんにちは" ;
035: } else {
036:   echo "こんばんわ" ;
037: }
038:
039: echo "<BR>\n" ;
040: echo $text;
041: ?&gt;
042: &lt;/body&gt;
043: &lt;/html&gt;
044: </pre>
045:
046: <h3>解説</h3>
047: <p>環境変数はプログラムなどが参照するデータベースで、その内容はサーバ自身の情報やアクセスしたク
クライアント
048: の情報などさまざまです。</p>
049: <p>31行目から37行目で使われている、if~elseは<a href="jyouken.html">条件分岐</a>といい、さまざま
な条件で違った結果を
050: 実行することをいいます。</p>
051: <p>表示したいPHP(16行目から20行目)はHTMLタグの外にだします。また呼び出したいPHP(27行目か
ら41行目)
052: はHTMLタグの中に入れなければなりません。</p>
```

053: <p><?php 最低 1 行分の PHP;?>というのを、HTML のどこでも書くことができます。何かを HTML の文として出力するには、

054: echo を使うと良いです。（「文字列を表示する」参照。）</p>

055:

056: <h3>TOP</h3>

057: </body>

058: </html>

```

001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <html lang="ja">
003: <head>
004: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
005: <meta http-equiv="Content-Style-Type" content="text/css">
006: <title>条件分岐とは？</title>
007: </head>
008: <h3>●条件分岐とは？●</h3>
009: <body>
010: <p> 変数$aがあり、$aが5より大きいとき「$aは5より大きい」と表示、そうでなければ「$aは5より小さい」と表示
011: させるような、さまざまな条件を基に違った結果を実行することを<B>条件分岐</B>といいます。</p>
012: <p>条件分岐には if~else という構文を使います。</p>
013:
014: <p> <B>if (</B>○○<B>) {</B> </p>
015: <p> △△ </p>
016: <p> <B>} else {</B> </p>
017: <p> □□ </p>
018: <p> <B>}</B> </p>
019:
020: <p> これは、「もし○○が正しければ△△を実行、○○が正しくなければ□□を実行する」という意味になります。
021: このように、ある式の結果が正しいことを TRUE (真)、正しくないことを FALSE (偽) といいます。</p>
022: <p> ここで先ほどの条件「変数$aがあり、$aが5より大きいときは"$aは5より大きい"と表示、そうでなければ
023: "$aは5より小さい"と表示させる」を日本語で記述してみると次のようになります。</p>
024:
025: <p> <B>if ($aが5より大きい) {</B> </p>
026: <p> <B>"$aは5より大きい"と表示</B> </p>
027: <p> <B>} else {</B> </p>
028: <p> <B>"$aは5より小さいと表示"</B> </p>
029: <p> <B>}</B> </p>
030:
031: <table border=4 width=500 >
032: <tr>
033: <td>例</td>
034: <td>意味</td>
035: <td>式の値</td>
036: </tr>
037: <tr>
038: <td>$a == $b</td>
039: <td>$a と $b は等しい</td>
040: <td>$a と $b が等しい場合に TRUE</td>
041: </tr>
042: <tr>
043: <td>$a != $b</td>
044: <td>$a と $b は等しくない</td>
045: <td>$a と $b が等しくない場合に TRUE</td>
046: </tr>
047: <tr>
048: <td>$a < $b</td>
049: <td>$a は $b より小さい</td>
050: <td>$a が $b より小さい場合に TRUE</td>
051: </tr>
052: <tr>
053: <td>$a > $b</td>
054: <td>$a は $b より大きい</td>

```

```

055: <td>$a が $b より大きい場合に TRUE</td>
056: </tr>
057: <tr>
058: <td>$a <= $b</td>
059: <td>$a は $b より小さいか等しい</td>
060: <td>$a が $b より小さいか等しい場合に TRUE</td>
061: </tr>
062: <tr>
063: <td>$a >= $b</td>
064: <td>$a は $b より大きい等しい</td>
065: <td>$a が $b より大きい等しい場合に TRUE</td>
066: </tr>
067: </table>
068: <p>「$a は 5 より大きい」と表示、「$a は 5 より小さい」と表示させるには、それぞれ echo 文で記述
069: しますので、これをスクリプトにすると次のようになります。</p>
070:
071: <p><B>if ($a > 5) {</B> </p>
072: <p><B>  echo '$a は 5 より大きい';</B> </p>
073: <p><B>} else {</B> </p>
074: <p><B>  echo '$a は 5 より小さい';</B> </p>
075: <p><B>}</B> </p>
076:
077: <h3><a href="sotsuron.html">TOP</a></h3>
078: </body>
079: </html>

```

スクリプトの中で、ある決まった処理を繰り返し行うには（ループ処理）、for 構文を使用します。それぞれの式の間は「;」（セミコロン）で区切りますので注意してください。

Kurikaeshi.html

```
001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <html lang="ja">
003: <head>
004: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
005: <meta http-equiv="Content-Style-Type" content="text/css">
006: <title>繰り返し(ループ)の処理</title>
007: </head>
008: <h1>繰り返し(ループ)の処理</h1>
009: <h2>for 構文</h2>
010: <h3>回答例</h3>
011: <body>
012:
013: <pre>
014: &lt;!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"
015: "http://www.w3.org/TR/html4/loose.dtd"
016: &lt;html&gt;
017: &lt;head&gt;
018: &lt;meta http-equiv="Content-Type" content="text/html; charset=euc-jp"&gt;
019: &lt;title&gt;test04&lt;/title&gt;
020: &lt;/head&gt;
021: &lt;body&gt;
022: &lt;?php
023: for ($i=1,$a=0,$i<=10;$i++) {
024:     echo "$a + $i = ";
025:     echo $a += $i;
026:     echo "<BR>\n" ;
027: }
028:
029: echo '$a は '. $a. ' です<BR>' ;
030: echo '$i は '. $i. ' です<BR>' ;
031: ?&gt;
032: &lt;/body&gt;
033: &lt;/html&gt;
034: </pre>
035:
036: <h3>解説</h3>
037: <p>・ループ処理の前に1の式が評価されます。その後、2の式が評価され、値が TRUE(真)の場合にループ処理が行われ、
038: 2の式の値が FALSE(偽)の場合はループ処理を終了します。</p>
039: <p>・それぞれの式の間は「;」（セミコロン）で区切ります。</p>
040:
041: <p>for 構文を使用することで、<a href="haireru.html">配列を便利に扱う</a>ことができます。</p>
042:
043: <h3><a href="sotsuron.html">TOP</a></h3>
044: </body>
045: </html>
```

Hairetu.html

```
001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <html lang="ja">
003: <head>
004: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
005: <meta http-equiv="Content-Style-Type" content="text/css">
006: <title>配列と f o r 構文</title>
007: </head>
008: <h2>配列と f o r 構文</h2>
009: <p> f o r 構文を配列することで、配列を便利に扱うことができます。次のスクリプトを実行し、 f o r 構文と配列の
010: 動作を確認してください。</p>
011:
012: <body>
013: <pre>
014: &lt;!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"
015: "http://www.w3.org/TR/html4/loose.dtd"
016: &lt;html&gt;
017: &lt;head&gt;
018: &lt;meta http-equiv="Content-Type" content="text/html; charset=euc-jp"&gt;
019: &lt;title&gt;test04&lt;/title&gt;
020: &lt;/head&gt;
021: &lt;body&gt;
022: &lt;?php
023: $seito = array (0 => "やまだたろう", 1 => "さとうじろう", 2 => "たなかさぶ
024: ろう", 3 => "はしもしろう", 4 => "なかむらごろう");
025:
026: echo '$seito の要素の数は ' . count($seito) . 'です&lt;BR&gt;&lt;BR&gt;';
027:
028: for ($i = 0;$i < count ($seito) ;$i++) {
029:   echo '$seito [' . $i . ']は ' . $seito [$i] . 'です&lt;BR&gt;';
030:   ?&gt;
031: &lt;/body&gt;
032: &lt;/html&gt;
033: </pre>
034:
035: ・ f o r 構文を使用することで、配列を便利に扱うことができます。</p>
036: <p>・ $seito の要素の数は5ですので、2の式の $i < count ($seito) は $i < 5 と展開されます。 $i の値
037: を配列の添字の値とすることで、配列の要素の値の一覧を表示させています。</p>
038: <p>・ ループ処理の後、 $i に 1 が足され、ループ処理が 5 回繰り返されると $i は 5 となり、2 の式の値が
FALSE(偽)と
039: なるので、ループの処理は終了します。
040: <h3><a href="sotsuron.html">TOP</a></h3>
041: </body>
042: </html>
```

演算子を使って、足し算の計算をするスクリプトを実行してみましょう。

Tashizan.html

```
001: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
002: <html lang="ja">
003: <head>
004: <meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp">
005: <meta http-equiv="Content-Style-Type" content="text/css">
006: <title>代数演算子</title>
007: </head>
008: <h1>足し算をする</h1>
009: <h2>代数演算子</h2>
010: <h3>回答例</h3>
011: <body>
012:
013: <pre>
014: &lt;!DOCTYPE html public "-//W3C//DTD HTML 4.01 Transitional//EN"
015: "http://www.w3.org/TR/html4/loose.dtd"
016: &lt;?php
017: $a = 10;
018: $b = 5;
019: ?&gt;
020: &lt;html&gt;
021: &lt;head&gt;
022: &lt;meta http-equiv="Content-Type" content="text/html; charset=euc-jp"&gt;
023: &lt;title&gt;test05&lt;/title&gt;
024: &lt;/head&gt;
025: &lt;body&gt;
026: &lt;?php
027: echo "$a + $b = ";
028: echo $a + $b . "&lt;BR&gt;";
029: ?&gt;
030:
031: &lt;/body&gt;
032: &lt;/html&gt;
033: </pre>
034:
035: <h3>解説</h3>
036: <p>・足し算や引き算など、数値と数値を計算するには代数演算子をつかいます。</p>
037: <p>・代数演算子には、おなじみの「+」や「-」のほかに次の演算子があります。</p>
038: <p>●代数演算子●</p>
039: <table border=4 width=500 >
040: <tr>
041: <td>例</td>
042: <td>名前</td>
043: <td>意味</td>
044: </tr>
045: <tr>
046: <td>$a+$b</td>
047: <td>加算</td>
048: <td>$a と $b の和</td>
049: </tr>
050: <tr>
051: <td>$a-$b</td>
052: <td>減算</td>
053: <td>$a と $b の差</td>
054: </tr>
055: <tr>
```

```
056: <td>$a*$b</td>
057: <td>乗算</td>
058: <td>$a と $b の積</td>
059: </tr>
060: <tr>
061: <td>$a/$b</td>
062: <td>除算</td>
063: <td>$a と $b の商</td>
064: </tr>
065: <tr>
066: <td>$a%$b</td>
067: <td>剰余</td>
068: <td>$a を $b で割った余り</td>
069: </tr>
070: </table>
071: <h3><a href="sotsuron.html">TOP</a></h3>
072: </body>
073: </html>
```